<u>In the Claims:</u>

Please amend the claims as indicated below:

Please cancel claims 5, 9, 10 and 23 without prejudice.

1. (currently amended) A method for identifying a runaway software agent operating in a computer system, said method comprising:

defining a time window for said agent;

receiving a current time signal;

determining a start time for said agent, said start time denoting when said agent began operating in said system;

determining if said time window is exceeded using said start time and said current time;

identifying said agent if said time window is exceeded, said identifying includes identifying said agent as a runaway agent; ~~and~~

<u>providing information about said agent to a user interface, without terminating said agent, wherein said user interface displays information regarding a plurality of runaway agents;</u>

<u>wherein said information about said agent includes an explanation of why said agent has exceeded said time window and a recommended course of action;</u>

<u>wherein said user interface further includes an execute solution display object that enables a user to cause said recommended course of action to be performed; and</u>

wherein said agent comprises a user developed task.

2. (original) The method of claim 1, wherein said time window includes an operating time limit for said agent.

3. (original) The method of claim 1, wherein said time window includes a starting event associated with said agent.

4. (original) The method of claim 1, further comprising:

      determining if another agent relies on said agent if said agent is a runaway agent; and

      notifying said another agent relying on said runaway agent.

5. (cancelled)

6. (original) The method of claim 1, further comprising:

      terminating said agent if said agent is identified.

7. (currently amended) The method of claim 1̲5̶, further comprising:

      allowing a user to specify a ranking for said agent within said plurality of runaway agents;

      wherein said ranking is responsive to a user specified severity associated with a type of agent behavior; and

      wherein said severity is associated with a corresponding time interval said agent is running behind schedule.

8. (currently amended) A method for identifying a runaway software agent operating in an HTTP environment on a computer system coupled to a communications network, said method comprising:

 identifying a thread associated with said HTTP environment using a thread identifier;

 initiating an agent from said thread;

 defining a time window associated with said agent;

 obtaining a start time for said agent on said thread;

 receiving a current time signal;

 determining if said time window is exceeded using said start time and said current time signal; ~~and~~

 identifying said agent as a runaway agent if said time window is exceeded;

 providing information about said runaway agent to a display device, without terminating said agent, wherein said display device displays information regarding a plurality of runaway agents;

 wherein said information about said runaway agent includes an explanation of why said runaway agent has exceeded said time window and a recommended course of action;

 wherein said display device further displays an execute solution display object that enables a user to cause said recommended course of action to be performed; and

 wherein said agent comprises a user developed task.

9. (cancelled)

10. (cancelled)

11. (currently amended) The method of claim 8~~10~~, further comprising:

ranking said agent within said plurality of other runaway agents, wherein said ranking is responsive to a user specified severity associated with a type of agent behavior, wherein said severity is associated with a corresponding time interval said agent is running behind schedule.

12. (previously presented) The method of claim 11, further comprising:

performing said recommended course of action on said runaway agent.

13. (previously presented) The method of claim 12, wherein said recommended course of action is performed, at least in part, by said user.

14. (previously presented) The method of claim 13, wherein said recommended course of action is performed, at least in part, by said system.

15. (previously presented) The method of claim 14, further comprising:

specifying a set of ranking criteria for said runaway agent using a user interface, wherein said set of ranking criteria includes said user specified severity associated with said type of agent behavior.

16. (original) The method of claim 15, wherein said ranking further comprises fatal, failure, high warning and low warning.

17. (currently amended) A method for managing a plurality of software agents operating in a computer system, said method comprising the steps of:

defining a time window for said plurality of agents;

identifying at least one of said plurality of agents exceeding said time window as a runaway agent to produce an identified set;

filtering said identified set according to predetermined filtering parameters;

ordering said agents within said identified set;

displaying said identified set, thereby managing the plurality of software agents operating in the computer system, said displaying including ; andproviding information about said agents in said identified set to a display device, without terminating said agents in said identified set, wherein said display device displays information regarding said agents in said identified set;

wherein said information about agents in said identified set includes an explanation of why each of said agents in said identified set has exceeded said time window and a recommended course of action;

wherein said display device further displays an execute solution display object that enables a user to cause said recommended course of action to be performed; and

wherein said software agents comprise user developed tasks.

18. (original) The method of claim 17, wherein said ordering step produces a rank ordered list of said agents in said identified set.

19. (original) The method of claim 17, wherein said filtering parameters are defined by a user.

20. (original) The method of claim 17, further comprising:

identifying members of said identified set having a thread associated therewith to produce a thread set.

21. (currently amended) A computer program product including a computer readable medium, said computer readable medium having machine-readable instructions stored thereon for instructing a processor to perform a method for identifying a software agent operating in a computer system, said machine-readable instructions comprising:

instructions for defining a time window associated with said agent;

instructions for receiving a current time signal;

instructions for determining a start time for said agent;

instructions for determining if said time window has been exceeded using said start time and said current time;

instructions for flagging said agent as a runaway agent if said time window is exceeded; ~~and~~

instructions for providing information about said agent to a user interface, without terminating said agent, wherein said user interface displays information regarding a plurality of runaway agents;

wherein said information about said agent includes an explanation of why said agent has exceeded said time window and a recommended course of action;

wherein said user interface further includes an execute solution display object that enables a user to cause said recommended course of action to be performed; and

wherein said runaway agent comprises a user developed task.

22. (previously presented) The computer program product of claim 21, wherein said time window includes an operating time limit for said agent.

23. (cancelled)

24. (previously presented) The computer program product of claim 21, said machine-readable instructions further comprising:

       instructions for terminating said agent if said agent is identified.